A New Tree-like Fuzzy Binary Support Vector Machines
For Optical Character Recognition

Guo-yun Zhang* a,b, Jing Zhang b
a College of Electrical and Information Engineering, Hunan University, Changsha 410082, China
b Department of Physics and Electronics Information, Hunan Institute of Science and Technology, Yueyang 414006, China

ABSTRACT

This paper proposes a new tree-like fuzzy binary support vector machines multi-class classifier (FBSVM) for the optical character recognition task. We construct this tree-like classifier by fusing of fuzzy clustering technique and support vector machine (SVM). In \(k\)-class task, the new classifier contains \(k-1\) SVM sub-classifiers, but the “one-against-one” method which is usually used contains \(k(k-1)/2\) sub-classifiers. This method also overcomes the drawback such as unclassifiable region that the “one-against-one” method has, and has a good classification performance. Furthermore, it needs less memory. By applying the new classifier to the real mail zipcode digits recognition task, the experimental results indicate that the FBSVM has a better recognition performance.

Keywords: Support vector machines, fuzzy clustering, multi-class classifier, optical character recognition

1. INTRODUCTION

The recognition of handwritten character has been an active research domain in recent years. In character recognition task, people face two fundamental problems, one is to segment the characters (letters and symbols), the other is to classify each of the characters[1][2]. Several approaches for character recognition have been proposed. Most of these algorithms achieve good performances in terms of correct recognition rate. But, in crucial real applications as automatic bankcheck reading systems or zipcode recognition systems, errors are very expensive to correct[3]. Recently, successful applications of support vector machines (SVMs) have been developed in character recognition task[3][4]. But, this task needs to construct a multi-class classifier. There are two construction methods in common use which are called as “one-against-one” and “one-against-all”, the “one-against-one” method has a better precision than the “one-against-all” method. Nevertheless, the “one-against-one” method has a disadvantage such as unclassifiable region. In \(k\)-class classifying task, it needs to construct \(k(k-1)/2\) sub-classifiers. Thus, too many sub-classifiers and memory[5][6] are required.

This paper constructs a tree-like fuzzy binary support vector machines multi-class classifier (FBSVM) by fusing of fuzzy clustering technique[7] and support vector machines[8][9] and with the idea of dichotomy. We apply this approach to the optical character recognition task and gain a better recognition performance. The paper is organized as follows. Basic concepts on SVMs are briefly presented in section 2. Two SVMs multi-class classifiers in common use are explained in section 3 and a new tree-like fuzzy binary support vector machines multi-class classifier (FBSVM) is proposed in section 4. Finally, the experiments which demonstrate the relevance of the FBSVM on a real zip code character recognition task and conclusions are gained in section 5 and 6, respectively.

2. SVM

Based on statistical learning theory (SLT), SVM[8][9] was progressed as a novel type of machine learning approach in the middle of nineties. It realizes the minimization of empirical risk and believable range through seeking the minimization of structural risk and in light of the best tradeoff between the complexity of the model (that is the learning precision of the specific training samples) and the learning capability (that is the capability of inerrancy recognizing the any samples) of the limited samples. Thereby, it can gain better generalization and statistical disciplinarian under the condition of less sample. Given training data \((x_1, y_1), \ldots, (x_n, y_n), x \in \mathbb{R}^d, y \in \{-1, +1\}\), \(n\) is the number of samples and \(d\) is the input dimension. There is a separating hyperplane \(f(x)\) which can separate the two groups of data completely with the data linear separable.
The solution of the optimal hyperplane can be considered as a quadratic programming problem. For the given training samples, the optimal weight $w$ and the offset $b$ is to find to minimize the constant function of the weight, that is

$$f(x) = (w \cdot x) + b = 0$$

(1)

When the training samples are not linear separable, there is a none-negative slack variable $\xi_i$, $i = 1, \cdots, l$. Thus, the optimal problem of the classification hyperplane is

$$\min \phi(x) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{l} \xi_i$$

$$s.t. \ y_i(w \cdot x_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0, \ i = 1, \cdots, l$$

(3)

where $C$ is penalty parameter which can reduce the number of training errors, the larger $C$ denotes the much penalty to error classifying.

Here, the optimization function $\phi(w)$ is quadratic form and the constraint condition is linear. So, it is a classical quadratic programming problem. We can employ Lagrangian multiplier approach to solve this problem:

$$L_p = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{l} \xi_i$$

$$- \sum_{i=1}^{l} \alpha_i \left[ y_i(x_i \cdot w + b) - 1 + \xi_i \right] - \sum_{i=1}^{l} \beta_i \xi_i$$

(4)

where $\alpha_i \geq 0, \beta_i \geq 0, \forall i$ are Lagrangian multiplier, the extreme point of $L_p$ is a saddle point, get the minimum $w=w^*, \ b=b^*$ ( $L$ to $w$ and $b$ ). The original problem can be translated into a dual quadratic programming problem:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

$$s.t \ y_i \alpha_i = 0, \ 0 \leq \alpha_i \leq C, i = 1, \cdots, l$$

(5)

By solving the quadratic programming problem, the corresponding optimal $\alpha^*$ is available and the classifying result is determined by the corresponding samples of the nonzero $\alpha^*$, those samples are called as support vector (SVs). When we test the testify sample $x$, which class $x$ belongs to is decided by the sign of the optimal decision function $f(x)$.

$$f(x) = \text{sgn}[\sum_{i=1}^{l} \alpha^* (x_i \cdot x) + b^*]$$

(6)

When the samples are nonlinear, the sample $x$ can be mapped to a high-dimensional feature space $H$ with a linear classifier in $H$. This is to transform $x$ into $H$ by a map function $\phi: R^d \rightarrow H \ , \ x \rightarrow \phi(x) = (\varphi_1(x), \varphi_2(x), \cdots, \varphi_l(x), x)^T$, where $\varphi_i(x)$ is real function.

According to the Mercer condition, the linear classification of a nonlinear transformation can be realized by adopting the varied inner product function $K(x, x_i)$ on the optimal classification plane. Here, $K(x_i, x_j)$ is called as kernel function and the dual problem is:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

(7)

and the corresponding optimal decision function is:

$$f(x) = \text{sgn}[\sum_{i=1}^{l} \alpha^* (x_i \cdot x) + b^*] = \text{sgn}[\sum_{i=1}^{l} \alpha^* K(x, x_i) + b^*]$$

(8)
At the present time, there are three popular kernel functions, i.e. \( \frac{1}{\sigma^2} \int e^{-\frac{|x-y|^2}{\sigma^2}} \) (RBF), \((x_i^T x_j + 1)^d\) (polynomial), \( \tanh(\zeta y + \beta) \) (hyperbolic tangent) etc., where \( \sigma, d, \zeta, \beta \) are kernel parameters.

### 3 COMMON SVM MULTI-CLASS CLASSIFICATION METHODS

#### 3.1 “one-against-one” method

**Definition 1.** In \( k \)-class classification task, we select the samples of two varied classes respectively to construct a SVM sub-classifier. Thereby, there are \( k(k-1)/2 \) sub-classifiers. We define this method as “one-against-one” method \([8]\).

To construct the SVM sub-classifier \( C_{ij} \) of the class \( i \) and \( j \), we select the training samples from the data samples set which belong to the class \( i \) and \( j \). Then, the samples which belong to the class \( i \) are labeled as positive class, while those samples which belong to the class \( j \) are labeled the negative class. We use those new training samples which we selected to construct a new sub-classifier \( C_{ij} \) and the corresponding samples set \( S_p \). There are different methods for doing the future testing after all \( k(k-1)/2 \) classifiers are constructed. After some tests, we can use the following voting strategy: if the sub-classifier \( C_{ij} \) says \( x \) is in the \( i \)-th class, then, the vote for the \( i \)-th class is added by one. Otherwise, the \( j \)-th is increased by one. Then we predict \( x \) that is in the class with the largest vote. The voting approach described above is also called the “Max Wins” strategy. In case that two classes have identical votes, thought it may not be a good strategy. That is there is an unclassifiable region.

#### 3.2. “one-against-all” method

**Definition 2.** In \( k \)-class classification task, the following approach to construct \( k \) SVM sub-classifiers is called “one-against-all” method \([8]\).

When we construct the \( i \)-th sub-classifier \( C_i \), we label those samples which belong to the \( i \)-th class as the positive class, the other samples are labeled the negative class. Then we reconstruct the training samples set \( S_i \) of the sub-classifier \( C_i \). When testing the input vector \( x \), we calculate the each output value of the \( k \) sub-classifiers and consider the corresponding class of the sub-classifier which has the maximum value of the decision function as the class which the testing sample belongs to.

### 4. TREE-LIKE FUZZY BINARY SVM MULTI-CLASS CLASSIFIER METHOD

This paper proposes a new tree-like fuzzy binary support vector machines multi-class classifier to improve the performance of the recognizer in \( k \)-class character recognition task.

#### 4.1. Tree-like fuzzy binary support vector machines multi-class classifier

**Definition 3.** The five-components group \( \text{FBSVM} = \langle U, S, F, \text{NewSet}, SVM \rangle \) satisfying the following conditions is called tree-like fuzzy binary support vector machines multi-class classifier (FBSVM).

1. \( F = \bigcup_{i=1}^{k} F_i \), \( F = \{(x_i, y_i), \cdots, (x_i, y_i)\}, \ F_i = \{(x_i, y_i), \cdots, (x_m, y_i)\}, \ x_j \in R^n, \ y_i \in \{1, 2, \cdots, k\}, \ \sum_{i=1}^{k} m_i = l \);

2. \( U = \{U_1, \cdots, U_I\} \), \( U_i \) is the clustering center of \( F_i \), \( U_i \in R^n \);

3. A two-components group \( \langle U, S \rangle \) constitutes binary tree, \( S \) is a set of some binary relation \( H \) in \( U \), \( (U_i, H) \) is a left subtree of the root \( r \), and \( (U_L, H_R) \) is a right subtree of the root \( r \), \( H_L \subset H \), \( H_R \subset H \).

4. \( \text{NewSet} = \{\text{NewSet}_1, \cdots, \text{NewSet}_{I-1}\}, \ \bigcup_{i=1}^{I-1} \text{NewSet}_i = F \). Where \( \text{NewSet}_i \) are the training samples of the \( i \)-th sub-classifier;

5. \( SVM = \{SVM_{c1}, \cdots, SVM_{c_{I+1}}\}, SVM_{ci} \) is a sub-classifier in the node \( i \).

Here, \( F \) is the training samples set and \( F_i \) is the \( k \) partitions of \( F \). According to the above definition, the learning process of the FBSVM is to construct \( k-1 \) SVM sub-classifier in fact. Firstly, this paper calculates the clustering centers \( U = \{U_1, \cdots, U_I\} \) of the samples of each class using fuzzy clustering technology and gains the root node. Then, the set \( U \) is
divided into two classes by clustering technology and the samples of the two classes are labeled positive class $P_1$ (left subtree) and negative class $N_1$ (right subtree) respectively. Thus, the new training set $NewSet_1$ and the first SVM sub-classifier $SVM_{c1}$ are reconstructed. And then those clustering centers belonging to $P_1$ are divided into two classes again by fuzzy clustering technology. The samples of the new two classes are labeled positive class $P_2$ and negative class $N_2$ and the second sub-classifier can be constructed with those samples. The same operating methods are employed to the negative class $N_1$. Then, the samples which belong to $P_2$ and $N_2$ are clustered into two classes and the third level sub-classifier can be constructed with the same method. Doing this in turn, till that there is only one clustering center point in each sub-class. Thereby, those sub-classifiers shape a tree structure showed in Fig. 1. one SVM sub-classifier would be constructed in each node.

![Fig.1. Structure of FBSVM](image)

**Theorem 1.** In $k$-class character recognition task, FBSVM contains $k-1$ SVM sub-classifier.

**Proof:** In $k$-class classification task, the number of the terminal node of FBSVM is $k$ because the degrees of all nodes are not more than 2 in binary tree.

Suppose the number of the node whose degree is 1 is $n_1$ and the number of the node whose degree is 2 is $n_2$, so the number of total nodes is:

$$n = k + n_1 + n_2$$  \hspace{1cm} (9)

In classification tree, there is a branch access to each node except the root node. Suppose $B$ is the number of total branches. Then $n = B + 1$. For these branches extend from the nodes which degree is 1 or 2, that is $n = n_1 + 2n_2$, that is

$$n = n_1 + 2n_2 + 1$$  \hspace{1cm} (10)

from equation (9) and (10), we can get

$$n_2 = k - 1$$  \hspace{1cm} (11)

because each node which degree is 2 represents a SVM sub-classifier, so FBSVM contains $k-1$ SVM sub-classifier.

In all $k-1$ SVM sub-classifiers constructed by FBSVM, the number of the training samples of the lower level sub-classifier (the child of the tree node) is less than that of the upper level sub-classifier (the parent of the tree node). So the needed computational time is decreasing. FBSVM divides the samples into two classes in each node position only according to the clustering center of each class samples. This avoids misclassification possibility of some bound sample in a certain extent.

### 4.2. Constructing SVM sub-classifier

When we construct the $i$th SVM sub-classifier $SVM_{ci}$, the samples of the left subtree are defined as positive $P_i$, and the samples of the right subtree are defined as negative $N_i$. $NewSet_i = P_i \cup N_i$. So the optimal problem of the $i$th SVM sub-classifier $SVM_{ci}$ is as follows,

$$\min \phi(x) = \frac{1}{2} \|w\|^2 + C \sum_{j=1}^{l_i} \xi_j$$  \hspace{1cm} (12)

s.t. $w_i K(x,x_j) + b_i \geq 1 - \xi_j$,  \hspace{1cm} $y_j \in P_i$

$w_i K(x,x_j) + b_i \leq \xi_j - 1$,  \hspace{1cm} $y_j \in N_i$

$$\xi_j \geq 0, \hspace{1cm} i = 1, \ldots, k - 1$$

where $l_i$ is the number of the elements in $NewSet_i$. 

Proc. of SPIE Vol. 5637 215
FBSVM needs to solve \( k \)-1 optimal problems like equation (12). So \( k \)-1 decision functions \( f_i(x), \ldots, f_{k-1}(x) \) can be gained. Each \( f_i(x) \) can be obtained by equation (8).

When we testify the samples, we can start the testing work from the root node sub-classifier \( SVM_{c_1} \) and discriminate its output whether it is positive or negative (denote with +1 and –1 respectively), then test the corresponding second level SVM sub-classifier according to the result in upper step. Doing this in turn, till the last level sub-classifier. Thereby, the class that the test samples belong to can be calculated.

### 4.3. FBSVM algorithm

The following is the description of the FBSVM training algorithm:

**Step1.** Calculating the fuzzy clustering center \( U = \{U_1, \ldots, U_k\} \) of each class of training samples, suppose there are \( k \) classes in total.

**Step2.** Dividing \( U \) into two classes \( P_1 \) and \( N_1 \) using fuzzy clustering technology. \( P_1 \cup N_1 = U \), \( P_1 \cap N_1 = \emptyset \), \( P_1 \subset U \), \( N_1 \subset U \);

**Step3.** Defining the training samples belong to \( P_1 \) as class +1, while defining the training samples belong to \( N_1 \) as –1. Re-combining the training samples into the set \( NewSet_1 \) to construct the SVM sub-classifier \( SVM_{c_1} \);

**Step4.** Clustering \( P_1 \) into two classes \( P_2 \) and \( N_2 \), and clustering \( N_1 \) into \( P_3 \) and \( N_3 \) where \( P_2 \cup N_2 = P_1 \), \( P_2 \cap N_2 = \emptyset \), \( P_2 \subset P_1 \), \( N_2 \subset P_1 \); \( P_3 \cup N_3 = N_1 \), \( P_3 \cap N_3 = \emptyset \), \( P_3 \subset N_1 \), \( N_3 \subset N_1 \);

**Step5.** Applying \( P_2 \) and \( N_2 \) to construct \( NewSet_2 \) and the SVM sub-classifier \( SVM_{c_2} \). Applying \( P_3 \) and \( N_3 \) to construct \( NewSet_3 \) and the SVM sub-classifier \( SVM_{c_3} \);

**Step6.** Repeating step 4 and step 5, till the set \( NewSet_{k-1} \) and the \( (k-1) \)th SVM sub-classifier \( SVM_{c_{k-1}} \) are constructed.

### 5. EXPERIMENTAL RESULTS

In this paper, the digit image set used for this experiment contains only segmented characters from real mail zipcodes and the source images are composed of 10 different characters, the numbers from 0 to 9. These images were obtained by scanning a laser printed text. We used this database to validate the proposed approach. The database contains 3000 handwritten samples and the testing set contains 1000 samples. Each digit is a 16×16 image represented as a 256-dimensional vector. We apply the FBSVM approach proposed in this paper and fuzzy clustering technology (FCM) to gain 10 clustering center \( C_1, \ldots, C_{10} \). Each clustering center corresponds to a kind of character \( C_1, \ldots, C_{10} \) correspond to digit character 0–9 respectively. Then 9 training sets are reconstructed, namely, \( NewSet = \{NewSet_1, \ldots, NewSet_{12}\} \) and 9 SVM sub-classifiers \( SVM_{c_1}, \ldots, SVM_{c_9} \). Fig. 2. shows the clustering tree structure. Here, \( T_i \) represents the ordinal number of the sub-classifier. \( T_1 = \{C_1, \ldots, C_{10}\} \), \( T_2 = \{C_2, C_3, C_6, C_{10}\} \), \( T_3 = \{C_2, C_4\} \), \( T_4 = \{C_4, C_{10}\} \), \( T_5 = \{C_1, C_5, C_8, C_9, C_{10}\} \), \( T_6 = \{C_1, C_6, C_8\} \), \( T_7 = \{C_3, C_5\} \), \( T_8 = \{C_3, C_7, C_9\} \), \( T_9 = \{C_5, C_9\} \). We have done the experiments with RBF kernel function and Poly kernel function respectively. For each sub-classifier, when we use the RBF kernel function, the penalty parameter \( C = 10 \), \( \sigma = 0.4 \), the stop error \( \epsilon = 0.001 \). While we use the Poly kernel function, the penalty parameter \( C = 100 \), \( d = 2 \), the stop error \( \epsilon = 0.001 \). Table 1. shows the recognition result. Where the classical RBF and SVM are the results in reference [5]. From the data in table 1., the FBSVM approach has less recognition error rate.

![Fig.2. Tree-like clustering structure of the clustering centers](image-url)
6. CONCLUSIONS

This paper proposes a tree-like fuzzy binary support vector machines multi-class classifier (FBSVM) by fusing of fuzzy clustering technique and support vector machines and using the concept of dichotomy. This approach constructs \(k\) SVM sub-classifier step by step based on calculating the clustering centers of each class and clustering the center dichotomously. Through the optical character recognition task, the following two conclusions are desirable: (1) whether the RBF kernel function or the Poly kernel function is used, the classification precision of the FBSVM is better than that of “one-against-one” method. This is mainly because the FBSVM approach contains rough classification and subdivision two procedures. Namely, the first step is use fuzzy clustering technology to complete the rough classification task and then use the SVM realize subdivision. (2) Theoretical proof FBSVM only contains \(k\) sub-classifiers. But the “one-against-one” needs \(k(k-1)/2\) sub-classifiers. So FBSVM needs less memory. In addition, FBSVM has not unclassifiable region because the tree-shape structure is applied.

ACKNOWLEDGMENTS

This work is partially supported by the science & technology research key project of the Ministry of Education of China ([2001]224) and the science & technology research project of the Department of Education of Hunan province (01C373).

REFERENCES


* gyzhang2283@sina.com, phone: 0086-0731-8714361 (13875800364).